NLG Evaluation: Let's open up the box

Chris Mellish
Computing Science
University of Aberdeen
Aberdeen AB24 3UE, UK
cmellish@csd.abdn.ac.uk

Donia Scott
Centre for Research in Computing
The Open University
Milton Keynes, MK7 6AA, UK
d.scott@open.ac.uk

Abstract

There is a spectrum of possible shared tasks that can be used to compare NLG systems and from which we can learn. A lot depends on how we set up the rules of these games. We argue that the most useful games are not necessarily the easiest ones to play.

The Lure of End-to-End Evaluation

Mellish and Dale (1998) discuss a number of different approaches to NLG system evaluation that had been used by 1998. Systems can be evaluated, for instance, in terms of accuracy, fluency or in their ability to support a human task. Independent of this is the question as to whether evaluation is *black box* or *glass box*, according to whether it results in an assessment only of the complete system or also of its contributing parts.

End-to-end evaluation is black box evaluation of complete NLG systems. It involves presenting systems with "naturally occurring" data and evaluating the language produced (according to accuracy, fluency, etc.). End-to-end evaluation is a tempting way to start doing NLG evaluation, because it imposes minimal constraints on the structure of the systems. Therefore as many people as possible can take part. This is important, because at the beginning critical mass is needed for things to "take off".

The Dangers of End-to-End Evaluation

Unfortunately there are dangers in using an end-to-end task as the basis of comparative NLG system evaluation:

- Danger of overfitting the task. The best systems may have little to say about language in general, but may encode elaborate stimulus-response type structures that work for this task only.
- Lack of generalisability. The best systems may have nothing to say about other NLG tasks. Or the way that systems are presented/ compared may prevent researchers in nearby areas from seeing the relevance of the techniques. So you may actually end up attracting *fewer* interested people.

Opening the box

End-to-end evaluation emphasises a "black box" approach that ignores what the NLG systems are doing inside. And yet we have some good ideas about the general tasks carried out in NLG (e.g., lexical choice, referring expression generation, aggregation) and it is at this level that we exchange knowledge at conferences and the field progresses independent of particular applications.

Opening the box for NLG evaluation would be analogous to the move in the MUC conferences from a unitary task to a set of much more structured subtasks. This was able to make MUC much more interesting to people involved in, for instance, named entity recognition and anaphora resolution. It also helped to bridge the large disconnect between 'success' in the MUC competition and 'progress' in the field of NLP.

Perhaps NLG evaluation could start simple and

progress in a similar way, moving in time from *application*-tasks to *NLG*-tasks. But without the significant funding that initiatives like MUC have had access to, it might well never make it beyond the first step.

How to Start?

How can we design evaluation tasks that stretch NLG systems in interesting ways? We need to have an agreement on which subtasks of NLG are of general interest and we need to have an agreement about what their inputs and outputs look like. This relies on a degree of theoretical convergence — something that the NLG field is not renowned for.

In this context, it is relevant to review whether RAGS (Mellish et al., 2006) might provide a good basis for defining tasks which would evaluate NLG systems, components and algorithms in a meaningful way.

RAGS

RAGS (Reference Architecture for Generation Systems) was an attempt to exploit previous ideas about common features between NLG systems in order to propose a reference architecture that would help researchers to share, modularise and evaluate NLG systems and their components without having to commit to particular theoretical approaches or implementational requirements. In practice, the project found that there was less agreement than expected among NLG researchers on the modules of an NLG system or the order of their running. On the other hand, there was reasonable agreement (at an abstract level) about the kinds of data that an NLG system needs to represent, in passing from some original non-linguistic input to a fully-formed linguistic description as its output.

RAGS took as a starting point eight commonly-agreed low-level NLG tasks (lexicalisation, aggregation, rhetorical structuring, referring expression generation, ordering, segmentation and centering/salience), and provided abstract type definitions for six different types of data representations (conceptual, rhetorical, document, semantic, syntactic and "quote"). It produced and made available sample implementations of the RAGS technology and complete implementations of RAGS systems, along with some sample datasets.

The final product of the RAGS project is undeniably incomplete, and the framework itself is difficult to use — both practically (e.g., many find the type descriptions hard to understand) and conceptually (one is forced to make hard decisions about the data at hand, answering questions such as "is this conceptual or semantic?").

Moving forward

There is a sense in which RAGS was slightly ahead of its time. Were we to start again, it would be more sensible to cast RAGS in terms of the Semantic Web (Berners-Lee et al., 2001). This would allow us to take advantage of the Web Ontology Language (OWL) (Antoniou and van Harmelen, 2003) and a great deal of technical infrastructure that has developed independently of, and in parallel to, RAGS.

We have begun to re-cast RAGS in terms of OWL, but this is still at an early stage. When complete, this work will help NLG researchers to use RAGS for the purpose for which it was intended: making it easier to create reusable data resources, communicate data between program modules, and allow modules (or at least their inputs and outputs) to be defined in a relatively formal way. This should make RAGS more useful for defining "glass box" evaluations of NLG systems.

This will not, of course, mean that evaluation would be an *easy* game to play; but, the game would be much more *meaningful*. And probably a lot more fun.

References

Grigoris Antoniou and Frank van Harmelen. 2003. Web Ontology Language: OWL. In S. Staab and R. Studer, editors, *Handbook on Ontologies in Information Systems*. Springer-Verlag.

T. Berners-Lee, J. Hendler, and O. Lassila. 2001. The semantic web. *Scientific American*, 284(5):35–43.

C. Mellish and R. Dale. 1998. Evaluation in the coatext of natural language generation. *Computer Speech and Language*, 12:349–372.

Chris Mellish, Donia Scott, Lynne Cahill, Daniel Paiva, Roger Evans, and Mike Reape. 2006. A reference architecture for natural language generation systems. *Natural Language Engineering*, 12(1):1–34, March.